一种树形模型链构建方法及其应用展望

刘志模[1]　刘　徽[3]　王晓山[2]　陈　振[2]
[1](西南计算机公司，重庆市，400060), [2,] (北京理工大学，北京市，100081)
[3](中国兵器工业计算机应用技术研究所，北京市，100089)

摘要：

[**目的**] 为运用数理逻辑模型论及推理方法来研究模型构建而提出一种模型组合方法。

[**方法**] 在数理逻辑模型论基础上，提出了模型受限并运算的方法，实现了模型在一定约束条件下的并运算，并进一步论述了子模型链、子模型树以及子模型网络的组建原理。

[**结果**] 论述了由小模型组建成大规模模型的新的可行途径和方法。即由较小的、成熟的的模型构建更大规模模型，或将较大规模模型分解为较小规模模型来实现。

[**局限**] 现只从数理逻辑模型论角度对模型构建提出了研究思路，进行了理论论述，尚未在具体模型实现中进行研究。

[**结论**] 为利用数理逻辑的知识表达和推理方法来研究模型构建给出了一种新的思路。

关键词：子模型；模型受限并运算；模型链构建；树形模型链；子模型网络

中图分类号：TP181

A Building Method About Tree Model Chain And The Application Prospect

Liu Zhimo[1]　Liu Hui[3]　Wang xiao shan[2]　Chen Zhen[2]
[1](XiNan Computer Corp.,ChongQing,400060,China),
[2](Beijing Institue of Technology, Beijing, 100081, China),
[3](China Ordnance Industry Computer Application Technology Research Institute, Beijing, 100089, China)

Abstract:

[Objective] A model combination method is proposed to study model construction using mathematical logic modeling and reasoning methods.

[Methods] On the basis of mathematical model theory, a method of constrained parallel operation of models is proposed, which realizes the parallel operation of models under certain constraint conditions, and further discusses the principles of building sub model chains, sub model trees, and sub model networks.

[Results] This paper discusses new feasible approaches and methods for building large-scale models from small models. That is, constructing larger scale models from smaller, mature models, or decomposing larger scale models into smaller scale models to achieve.

[Limitations] At present, research ideas and theoretical discussions on model construction have only been proposed from the perspective of mathematical logic model theory, and have not yet been studied in specific model implementation.

[Conclusions] A new approach has been proposed to utilize the knowledge expression and reasoning methods of mathematical logic to study the construction of models.
Keywords: sub-model；limited union operation of models； model chain construction；tree model chain；sub-model network

1. 引言

随着芯片、计算机、人工智能算法等技术的快速进步，人工智能进入了飞速发展时期。Trangsformer 模型采用自注意力机制和多头并行处理结构[1]，在超级计算机上能实现数十亿参数在数十亿未标注数据上进行自监督学习，使模型在自然语言处理方面取得重大突破。之后，BERT、ChatGPT-3、ChatGPT-4 等大模型相继诞生[2][3]，不仅在自然语言处理方面取得更大进步，而且实现了文本、编程、图像、语音、视频、音乐等多模态信息的处理和转换。以此为基础的生成式人工智能技术亦取得了瞩目进展。

在 Trangsformer 基础上发展起来的大模型现在已能调节上千亿参数进行深度学习，成为当下人工智能领域的主流方向。但另一方面亦应看到，这些大模型通过调节内部参数来进行学习，模型具有高度复杂性和非线性，存在可解释性和复现困难的问题[4]。它们的模型原理是什么，能否直接用于其他智能体设备的研发，这些问题都是人们关心，但直接从当前的大模型获取知识又相当困难的问题。人们的一种期望是，通过人工智能领域的研发活动，一些优秀的、成熟的模型和算法能够借鉴或直接用于新的智能体设备或模型、算法的研究。

为了在新建模型中借鉴和使用成熟的模型和知识，需对模型的含义和范围作深入的探究。在具体的科技领域中，所涉模型都是针对具体对象而建立的实体模型，模型所涉及的理论、算法、计算公式等都是针对具体对象而建立的。提及模型，人们也会联想到数理逻辑的模型论，其中对模型的概念有严格的定义，而且建立了一套严格的理论体系，如能将其运用到当前人工智能模型的研究，那是再好不过了。但数理逻辑模型论中所说的模型是一种满足一定逻辑规则的数学结构，不是涉及具体对象的实体模型，而是更抽象的结构模型。这种模型是由形式语言通过解释（映射）所定义产生的，其中包含了若干函数和算式。由于这种模型通过形式语言严格进行了定义，满足形式语言的推理规则，由它产生的逻辑推理结论应该适用于一般的实体模型推理。因此，研究数理逻辑模型对智能体的研发有着重要现实意义。

就模型而言，人们期望的一种情景是：可以由若干简单模型组合成更大的模型；或反过来，复杂的较大模型可以分解为若干较小的简单模型来求解。这样，利用相对简单、成熟的模型能够组合产生复杂的、较大的模型。这对人工智能系统、独立的智能体的研究来说有重要的意义。要实现模型的组合就必须要求进行模型的并运算，然而在数理逻辑模型论中，一般情况下的模型并运算是不满足运算封闭要求的[5]，因此无法定义，也就无法实现模型的并运算。现有模型论理论对此未能进行深入研究，这是一种缺失。鉴于人工智能领域的实际需要，本文将针对这一情况进行探究，采取增加约束条件的办法来实现模型的并运算，进而实现由小模型组合成更大模型的愿景。

2. 一阶逻辑结构及相关性质

2.1 一阶逻辑结构与模型

对于一阶逻辑结构与模型的有关定义与论述，详见参考文献[5][6][7][8]。这里列出本文中将用到的一些结论和约定。

(1) 一阶语言符号系统

本文所涉一阶逻辑要素，使用以下符号系统表示：

论域：研究对象所构成的集合，一般以 A, M ,… 表示；

个体记号：用来指定论域中元素的集合，一般以 $C=\{a, b, c, a_1, b_1, c_1, \ldots\}$表示；

关系记号：用来指定论域中元素之间的关系的集合，一般以 $R=\{H, P, r, p, r_1, p_1, \ldots\}$ 表示，对特殊的相等关系，以 $\approx$ 表示；

函数记号：用来指定论域中元素所构成的函数的集合，一般以 $F=\{G, f, g, f_1, g_1, \ldots\}$ 表示；

变元记号集合：$\{x, y, \ldots, u, v, \ldots\}$；

逻辑运算符集合：$\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$；

量词记号集合：{ ∀，∃ }；

标点记号集合：{ (，) ，， }。

以上记号组成一阶形式语言 $\mathcal{L}$：

$$\mathcal{L}=\{a, \ldots, R, \ldots, \approx, f, \ldots, x, \ldots, u, \ldots, \neg, \ldots, \forall, \exists, (\ ,\ )\ ,\ ,\ \}\text{。}$$

其中，$\{\approx, x, \ldots, u, \ldots, \neg, \ldots, \forall, \exists, (\ ,\ )\ ,\ ,\ \}$ 是不同系统中的共有部分，一般在特定系统中就不再列出。

定义 2.1： 设

$C=\{c_1, c_2, \ldots, c_\kappa\}$ 是一些个体记号组成的集合，

$R=\{r_1, r_2, \ldots, r_\lambda\}$ 是一些关系记号组成的集合，

$F=\{f_1, f_2, \ldots, f_\mu\}$ 是一些函数记号组成的集合，

则 $\mathcal{L}=\{C, R, F\}=\{c_1, c_2, \ldots, c_\kappa, r_1, r_2, \ldots, r_\lambda, f_1, f_2, \ldots, f_\mu\}$构成一个特定的语言，其中，$C$、$R$、$F$ 的基数 κ 、λ 、μ 可以是有限或者无限基数。

本文中所涉及的一阶逻辑的项、公式、语句等的定义和形成规则，以及变元在论域中的取值解释等同一般文献。

(2) 一阶语言的结构与模型

定义 2.2： 设$\mathcal{L}$为一阶语言。一个 $\mathcal{L}$-结构是一个二元组 $\mathcal{M}=(M, I)$，其中M为非空集合，称为$\mathcal{M}$ 的论域，I 为定义在$\mathcal{L}$上满足如下条件的映射(称 I为**解释**)：

1） I把$\mathcal{L}$中的每个常项符 c 解释为M中的元素，即 $I(c) \in M$ ；

2） I把 $\mathcal{L}$中的每个 n 元关系符 P解释为M 上的 n 元关系，即 $I(P) \subseteq M^n$ ；

3） I把$\mathcal{L}$中的每个 n 元函数符f解释为M上的 n 元函数，即 $I(f)$为从 M^n 到M的映射，$I(f)$：$M^n \rightarrow M$ 。

常把 $I(c)$、$I(P)$、$I(f)$ 分别记为 c^M、P^M、f^M 。则对如上的 $\mathcal{L}$-结构，有 $c^M \in M$，$P^M \subseteq M^n$，f^M：$M^n \to M$。

在一些资料[5][8]中，将二元组 $\mathcal{M} = (M, I)$看成为模型是针对 $\mathcal{L}$ 中的公式 A 或公式集 Σ 而定义的, 即当 $\mathcal{M} \vDash$ A(或 $\mathcal{M} \vDash \Sigma$)时, 定义 $\mathcal{M}$ 是 A(或 Σ)的模型。而在其它资料[6]中，把二元组 (M, I)看成为对语言 $\mathcal{L} = \{C, R, F\}$ 在 M 上的一种解释，当该二元组已把 $\mathcal{L}$ 解释为$\{\bar{C}, \bar{R}, \bar{F}\}$后, 则把 $\mathfrak{A} = (M, \bar{C}, \bar{R}, \bar{F})$ 叫做语言 $\mathcal{L}$ 的一个模型，M叫做 $\mathfrak{A}$ 的论域。

从结构上看，二者实际上是一致的。本文只是从数据结构上来探讨逻辑模型的问题，因此，采用[6]的方法，将把论域上实现解释后的结构看成为语言的模型。

定义 2.3： 设语言 $\mathcal{L} = \{C, R, F\}$。如果二元组$(M, I)$将 $\mathcal{L}$ 的个体、关系和函数记号在 M 中对应地作出了解释$\{\bar{C}, \bar{R}, \bar{F}\}$, 则把 $\mathcal{M} = (M, \bar{C}, \bar{R}, \bar{F})$称作 $\mathcal{L}$ 的**模型**，M 称作 $\mathcal{M}$ 的论域[6]。

本文以下讨论中，在提及语言 $\mathcal{L} = \{C, R, F\}$ 的模型时，已默认将 $\mathcal{L}$ 的个体、关系和函数记号在 M 中对应地作出了解释，为方便计，亦将 $\mathcal{M} = (M, C, R, F)$称作 $\mathcal{L}$ 的模型。

由定义 2.3 可看出，$\mathcal{L}$ 的模型 $\mathcal{M} = (M, C, R, F)$实质上是一种结构模型，可以将其看成为一个逻辑整体，其中含有若干函数和若干谓词确定的逻辑表达式。数理逻辑模型论重点研究了模型间的关系和推理，这对具体模型的研究有重要的应用价值。

2.2 逻辑结构的有关关系

为了本文进一步深入讨论，现将已有的几种逻辑结构的定义及关系列出[5][6]。

(1) 子模型：

定义 2.4： 设 $\mathcal{M}$，$\mathcal{N}$为同一语言 $\mathcal{L} = \{C, R, F\}$的两个模型。称 $\mathcal{M}$ 是 $\mathcal{N}$ 的子模型，记作 $\mathcal{M} \subseteq \mathcal{N}$，如果下列条件成立：

1） $M \subseteq N$；

2） 对任意常项符 $c \in \mathcal{L}$，有 $c^M = c^N$；

3） 对任意 n-元关系符 $R \in \mathcal{L}$，及任意 $a_1, \ldots, a_n \in M$, 都有

$R^M(a_1, \ldots, a_n)$ 当且仅当 $R^N(a_1, \ldots, a_n)$；

4） 对任意 n-元函数符 $f \in \mathcal{L}$，及任意 $a_1, \ldots, a_n \in M$, 都有

$f^M(a_1, \ldots, a_n) = f^N(a_1, \ldots, a_n)$。

对第 2）条，实际隐含 $c^M \in M$。

(2) 模型链：

定义 2.5： 设语言 $\mathcal{L} = \{C, R, F\}$。如果有一系列 $\mathcal{L}$ 的模型

$$\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_\beta, \cdots \qquad \beta < \alpha \text{（}\alpha\text{为序数）} \tag{2.1}$$

满足下列条件

$$\mathcal{M}_\beta \subseteq \mathcal{M}_\gamma \qquad \beta < \gamma < \alpha \tag{2.2}$$

则式(2.1)的诸模型就构成一条模型的升链：

$$\mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \cdots \subseteq \mathcal{M}_\beta \subset \cdots \quad \beta < \alpha \tag{2.3}$$

3. 模型的交、并运算

模型的交、并运算是模型结构拓展的基础，现对其运算规则进行约定。

3.1 模型的交运算

定义3.1： 设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\boldsymbol{\mathcal{L}}=\{C,R,F\}$ 的两个模型，其中$\mathcal{M}_1 = (M_1,C_1,R_1,F_1)$，$\mathcal{M}_2 = (M_2,C_2,R_2,F_2)$。定义 $\mathcal{M}_0 = \mathcal{M}_1 \cap \mathcal{M}_2 = (M_0,C_0,R_0,F_0)$为二者的交运算，满足下列条件：

1) 论域 $M_0 = M_1 \cap M_2$；
2) 对任意常项符 $c \in \boldsymbol{\mathcal{L}}$，有 $c^{M_0} = c^{M_1}$ 且 c^{M_2}；
3) 对任意 n-元关系符 $R \in \boldsymbol{\mathcal{L}}$，及任意 $a_1,\ldots,a_n \in M_0$，都有 $R^{M_0}(a_1,\ldots,a_n)$ 当且仅当（$R^{M_1}(a_1,\ldots,a_n)$ 且 $R^{M_2}(a_1,\ldots,a_n)$）；
4) 对任意 n-元函数符 $f \in \boldsymbol{\mathcal{L}}$，及任意 $a_1,\ldots,a_n \in M_0$，都有$f^{M_0}(a_1,\ldots,a_n)$ =（$f^{M_1}(a_1,\ldots,a_n)$ 且 $f^{M_2}(a_1,\ldots,a_n)$）。

在定义 3.1 中，由于论域 $M_0 = M_1 \cap M_2$ 的限定，对第 2)条，可得 $c^{M_0} = c^{M_1} = c^{M_2}$，于是 $C_0 = C_1 \cap C_2$；对第 3)条，可得 $R_0 = R_1 \cap R_2$；对第 4)条，排除了 $f^{M_1}(a_1,\ldots,a_n) \neq f^{M_2}(a_1,\ldots,a_n)$的可能，于是有$f^{M_0}(a_1,\ldots,a_n) = f^{M_1}(a_1,\ldots,a_n) = f^{M_2}(a_1,\ldots,a_n)$，可得 $F_0 = F_1 \cap F_2$。

由上可以看出，模型的交运算对常项符、关系和函数都是封闭的。

3.2 模型的并运算

与模型的交运算相比，模型的并运算情况要复杂得多。设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\boldsymbol{\mathcal{L}}=\{C,R,F\}$ 的两个模型，其中$\mathcal{M}_1 = (M_1,C_1,R_1,F_1)$，$\mathcal{M}_2 = (M_2,C_2,R_2,F_2)$。如果定义 $\mathcal{M}_0 = \mathcal{M}_1 \cup \mathcal{M}_2 = (M_0,C_0,R_0,F_0)$，则首先要求

$$M_0 = M_1 \cup M_2 \tag{3.1}$$

对某一 n-元函数 $f(x_1,\ldots,x_n) \in \boldsymbol{\mathcal{L}}$，它在 $\mathcal{M}_1$ 和某 $\mathcal{M}_2$ 中都得到解释，对取值组合 $\{(a_1,\ldots,a_n) | a_i \in M_0\}$，有以下几种情况：

1) $\{(a_1,\ldots,a_n) | a_i \in M_0, \forall a_i \in M_1\}$，这时 $f^{M_1}(a_1,\ldots,a_n) \in M_1 \Rightarrow f^{M_0}(a_1,\ldots,a_n) \in M_0$，运算封闭；
2) $\{(a_1,\ldots,a_n) | a_i \in M_0, \forall a_i \in M_2\}$，这时 $f^{M_2}(a_1,\ldots,a_n) \in M_2 \Rightarrow f^{M_0}(a_1,\ldots,a_n) \in M_0$，运算封闭；
3) 存在$\{(a_1,\ldots,a_i,\ldots,a_j,\ldots,a_n) | a_i, a_j \in M_0, \exists (a_i \in M_1 \text{且} a_j \in M_2 \text{且} a_j \notin M_1 \cap M_2)\}$，在这种情况下，$f^{M_0}(a_1,\ldots,a_i,\ldots,a_j,\ldots,a_n)$的值无法确定是否在 $M_0 = \mathcal{M}_1 \cup \mathcal{M}_2$ 中，造成函数不封闭。

例3.1：（模型并运算例） 设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\boldsymbol{\mathcal{L}}=\{C,R,F\}$ 的两个模型，其中

$\mathcal{M}_1=(M_1,C_1,R_1,F_1)$，$\mathcal{M}_2=(M_2,C_2,R_2,F_2)$。

假设 $\mathcal{M}_1$ 是一个二值逻辑系统的模型。其论域 M_1 ={0, 1}, 0 为最小元, 1 为最大元。函数 $f\in F_1$ 为 $\mathcal{M}_1$ 中定义的一个 n 元二值逻辑函数 $f(x_1,x_2,\dots,x_n)$ ，其中 $x_1,x_2,\dots,x_n$ 在 M_1 ={0, 1}上取值，f 是定义在 M_1 上的映射 f ：$M_1^n\to M_1$, 即函数 f 在 M_1 ={0, 1}上取值。

设 $\mathcal{M}_2$ 是一个三值逻辑系统的模型，其论域 M_2 ={0, u, 1}，0 为最小元，u 为中间元，1 为最大元。函数 $g\in F_2$ 为 $\mathcal{M}_2$ 中定义的一个 m 元三值逻辑函数 $g(y_1,y_2,\dots,y_m)$，其中 $y_1,y_2,\dots,y_m$ 在 M_2 ={0, u, 1}上取值，g 是定义在 M_2 上的映射 g ：$M_2^m\to M_2$, 即函数 g 在 M_2 ={0, u, 1}上取值。

现考察 $\mathcal{M}_1,\mathcal{M}_2$ 的并模型 $\mathcal{M}_0=\mathcal{M}_1\cup\mathcal{M}_2=(M_0,C_0,R_0,F_0)$，其中论域 $M_0=M_1\cup M_2$，函数域 $F_0=F_1\cup F_2$，即有 M_0={0, u, 1}，$f\in F_0$ ，$g\in F_0$ 。在这种情况下，显然 g 是有定义的，在并模型 $\mathcal{M}_0$ 中能求出函数 g 的值。但对函数 $f=f(x_1,x_2,\dots,x_n)$, $x_1,x_2,\dots,x_n$ 在原来的定义域{0, 1}内取值时 $f(x_1,x_2,\dots,x_n)$ 是成立的，而在新的定义域 M_0={0, u, 1}内，$f(x_1,x_2,\dots,x_n)$ 是否成立或满足原来的逻辑关系，将无法确定，造成函数不封闭，并模型不成立。更具体点, 假定 $f=x_1\bar{x}_2+\bar{x}_1x_2$ 是某系统中的一个二值“异或”逻辑控制函数，当实现并模型后，函数定义域扩大为{0, u, 1}，则在并模型系统中，控制函数 f 将不能正常动作。

由于存在函数不封闭的情况，一般就无法定义模型的并运算。如果要定义模型的并运算，则必须对模型条件作出某些限定才行。

定义3.2：（**模型受限并运算**）设模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 为一阶语言 $\mathcal{L}$={C, R, F }的两个模型，其中$\mathcal{M}_1=(M_1,C_1,R_1,F_1)$，$\mathcal{M}_2=(M_2,C_2,R_2,F_2)$。定义 $\mathcal{M}_0=\mathcal{M}_1\cup_{\lim}\mathcal{M}_2=(M_0,C_0,R_0,F_0)$ 为模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 的受限并运算，如果满足如下条件：

1) 论域 $M_0=M_1\cup M_2$；
2) 对任意常项符 $\mathrm{c}\in\boldsymbol{\mathcal{L}}$, 如果 $\mathrm{c}^{M_1}=\mathrm{c}^{M_0}$, 则 $\mathrm{c}\in C_1$ 且 $\mathrm{c}\in C_0$; 如果 $\mathrm{c}^{M_2}=\mathrm{c}^{M_0}$, 则 $\mathrm{c}\in C_2$ 且 $\mathrm{c}\in C_0$；
3) 对任意 n-元关系符 $R\in\boldsymbol{\mathcal{L}}$，及任意 $a_1,\dots,a_n$，如果 $a_1,\dots,a_n\in M_1$, 则有 $R^{M_1}(a_1,\dots,a_n)$ 当且仅当 $R^{M_0}(a_1,\dots,a_n)$；如果 $a_1,\dots,a_n\in M_2$，则有 $R^{M_2}(a_1,\dots,a_n)$ 当且仅当 $R^{M_0}(a_1,\dots,a_n)$。
4) 对任意 n-元函数符 $f\in\boldsymbol{\mathcal{L}}$，及任意 $a_1,\dots,a_n$，如果 $a_1,\dots,a_n\in M_1$, 则有 $\boldsymbol{f}^{M_1}(a_1,\dots,a_n)=f^{M_0}(a_1,\dots,a_n)$；如果 $a_1,\dots,a_n\in M_2$, 则有 $f^{M_2}(a_1,\dots,a_n)=f^{M_0}(a_1,\dots,a_n)$。
5) 对任意 n-元变量取值组合 $a_1,\dots,a_i,\dots,a_n\in M_0$，其取值范围：只存在 $(\forall a_i\in M_1)\vee(\forall a_i\in M_2)$ 的情形(包括 $a_i\in M_1\cap M_2$)。

以上条件称作**模型受限并运算条件**。

注意，定义 3.2 的第 5 款排除了$\{(a_1,\dots,a_i,\dots,a_j,\dots,a_n)|\, a_i,a_j\in M_0$，$\exists(a_i\in M_1$ 且 $a_j\in M_2$ 且 $a_j\notin M_1\cap M_2)\}$的情形，即 n-元变量取值范围为：

$\{(a_1,\dots,a_i,\dots,a_j,\dots,a_n) \mid a_i,a_j \in M_0\ ,\ \neg(\exists(a_i \in M_1 \wedge a_j \in M_2 \wedge a_j \notin M_1 \cap M_2))\}$。

引理 3.1： 模型受限并运算对关系和函数是封闭的。

引理 3.1 的结论是显然的。因为 n-元变量取值组合中任意变量的取值不会出现一些变量在 M_1 中取值，同时一些变量又在 M_2 且不在 $(M_1 \cap M_2)$ 中取值的情况，因而这种取值组合所确定的关系和函数是值可确定的，且均在并模型的论域范围内。

在定义 3.2 中，由于有了第 5)条的限制，这样定义的并模型就不会出现关系和函数不封闭的情况，因而是可行的。另一方面，这样的限制在实际应用中亦是合理的，比如在模型 $\mathcal{M}_1$ 中引用模型 $\mathcal{M}_2$ 时，模型 $\mathcal{M}_2$ 运行时的论域一定是该模型自身定义和成立的数据范围。一般而言，**不受限的模型并运算体现了从形式语言的语法角度去表达的思想，而受限模型并运算则是侧重于从语义角度去表达和实现，更接近于具体模型的研究**。

4. 模型结构的拓展

4.1 子模型的并运算

在已知两个模型是某个模型的子模型的前提下，是可以定义它们的并运算的。在实际应用中，为了保证子模型使用的正确性和完整性，也必须对子模型的引用作出某些限制。采用定义 3.2 的办法可以实现子模型的并运算。

定义 4.1： 设一阶语言 $\mathcal{L}=\{C,R,F\}$ 的模型 $\mathcal{M}_1=(M_1,C_1,R_1,F_1)$，$\mathcal{M}_2=(M_2,C_2,R_2,F_2)$ 是模型 $\mathcal{M}_0=(M_0,C_0,R_0,F_0)$ 的子模型，$\mathcal{M}_1 \subseteq \mathcal{M}_0$，$\mathcal{M}_2 \subseteq \mathcal{M}_0$。定义 $\mathcal{M}_0'=(M_0',C_0',R_0',F_0') = \mathcal{M}_1 \cup_{sub} \mathcal{M}_2$ 为子模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 的并模型，如果满足如下条件：

1） 论域 $M_1 \cup M_2 = M_0' \subseteq M_0$；

2） $\mathcal{M}_1$，$\mathcal{M}_2$ 的并运算满足定义 3.2 中的模型受限并运算条件。

定理 4.1： 定义 4.1 对子模型并运算的限定是可行的。

证明： 设一阶语言 $\mathcal{L}=\{C,R,F\}$ 的模 $\mathcal{M}_1=(M_1,C_1,R_1,F_1)$，$\mathcal{M}_2=(M_2,C_2,R_2,F_2)$ 是模型 $\mathcal{M}_0=(M_0,C_0,R_0,F_0)$ 的子模型。根据定义 2.4，有 $\mathcal{M}_1 \subseteq \mathcal{M}_0$ 及 $\mathcal{M}_2 \subseteq \mathcal{M}_0$，可得以下结果：

1） 对论域有 $M_1 \subseteq M_0$，$M_2 \subseteq M_0$，同时有 $M_1 \cup M_2 = M_0' \subseteq M_0$。

2） 对任意常项符 $c \in \mathcal{L}$，依据子模型定义，如果 $c^{M_1}=c^{M_0}$，则 $c \in C_1$ 且 $c \in C_0'$，$c \in C_0$；如果 $c^{M_2}=c^{M_0}$，则 $c \in C_2$ 且 $c \in C_0'$，$c \in C_0$。

3） 对任意 n-元关系符 $R \in \mathcal{L}$，及任意 $a_1,\dots,a_n$，如果 $a_1,\dots,a_n \in M_1$，则有 $a_1,\dots,a_n \in M_0'$，$a_1,\dots,a_n \in M_0$，$R^{M_1}(a_1,\dots,a_n)$ 当且仅当 $R^{M_0'}(a_1,\dots,a_n)$ 当且仅当 $R^{M_0}(a_1,\dots,a_n)$；如果 $a_1,\dots,a_n \in M_2$，则有 $a_1,\dots,a_n \in M_0'$，$a_1,\dots,a_n \in M_0$，$R^{M_2}(a_1,\dots,a_n)$ 当且仅当 $R^{M_0'}(a_1,\dots,a_n)$ 当且仅当 $R^{M_0}(a_1,\dots,a_n)$。

4） 对任意 n-元函数符 $f \in \mathcal{L}$，及任意 $a_1,\dots,a_n$，如果 $a_1,\dots,a_n \in M_1$，则有 $a_1,\dots,a_n \in M_0'$，$a_1,\dots,a_n \in M_0$，$\boldsymbol{f}^{M_1}(a_1,\dots,a_n) = f^{M_0'}(a_1,\dots,a_n) = f^{M_0}(a_1,\dots,a_n)$；如果 $a_1,\dots,a_n \in M_2$，则有 $a_1,\dots,a_n \in M_0'$，$a_1,\dots,a_n \in M_0$，$f^{M_2}(a_1,\dots,a_n) = f^{M_0'}(a_1,\dots,a_n) = f^{M_0}(a_1,\dots,a_n)$。

5） 对任意 n-元变量取值组合 $a_1, \ldots, a_i, \ldots, a_n \in M_0'$，其取值范围：只存在 $(\forall a_i \in M_1) \vee (\forall a_i \in M_2)$ 的情形(包括 $a_i \in M_1 \cap M_2$)，故不存在关系和函数不封闭的情形。

即是说，按照定义 4.1 得到的两个子模型的并模型是有确定意义的，故可行。■

定理 4.1 为我们在模型中**引用**其它子模型提供了依据。有以下推论：

推论4. 1： 定义 4.1 中，子模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$ 是模型 $\mathcal{M}_0 = (M_0, C_0, R_0, F_0)$ 的子模型，$\mathcal{M}_0' = (M_0', C_0', R_0', F_0') = \mathcal{M}_1 \cup_{\text{sub}} \mathcal{M}_2$ 为子模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 的受限并模型，则 $\mathcal{M}_1$，$\mathcal{M}_2$ 是 $\mathcal{M}_0'$ 的子模型，同时，$\mathcal{M}_0'$ 也是 $\mathcal{M}_0$ 的子模型。

根据模型受限并运算与子模型的定义，容易看出推论 4.1 成立。

推论4. 2： 设模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$，$\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$ 是一阶语言 $\boldsymbol{\mathcal{L}} = \{C, R, F\}$ 的两个模型，$\mathcal{M}_0 = (M_0, C_0, R_0, F_0)$ 是 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 按定义 3.2 模型受限并运算得到的并模型，$\mathcal{M}_0 = \mathcal{M}_1 \cup_{\lim} \mathcal{M}_2$，则 $\mathcal{M}_1$，$\mathcal{M}_2$ 均是 $\mathcal{M}_0$ 的子模型，即有 $\mathcal{M}_1 \subseteq \mathcal{M}_0$，$\mathcal{M}_2 \subseteq \mathcal{M}_0$。

根据定义 3.2 和子模型定义，不难验证推论 4.2 成立。实际上，定义 4.1 确定的子模型并运算 $\cup_{sub}$ 是由受限模型并运算 $\cup_{\lim}$ 所限定的，而 $\cup_{sub}$ 更能体现出并运算结果的子模型特征，根据推论 4.2，本文之后对任意模型的受限并运算均使用符号 $\cup_{sub}$ 了。

4.2 子模型的模型链

在定义 2.5 中，对模型的升链结构作出了规定。

在实际应用中，我们常常会用到模型的降链结构，以便于对其它成熟模型的引用；同时，模型的长度也可能是有限的，尽管其长度 γ 会很大。

定义 4.2： 设语言 $\mathcal{L} = \{C, R, F\}$ 的一系列模型为 $\mathcal{M}_\Gamma$：$\mathcal{M}_\gamma$，$\mathcal{M}_{\gamma-1}$，… ，$\mathcal{M}_1$，$\mathcal{M}_0$ ，其中 $\mathcal{M}_{i-1}$ 是 $\mathcal{M}_i$ 的子模型，即有 $\mathcal{M}_{i-1} \subseteq \mathcal{M}_i$ ($\gamma \geq i \geq 1$， $\gamma \in I$)。这一系列模型构成子模型升链：

$$\mathcal{M}_0 \subseteq \mathcal{M}_1 \subseteq \cdots \subseteq \mathcal{M}_\gamma \tag{4.1}$$

其中，子模型的个数 $\gamma + 1$ 称为模型链的长度，$\mathcal{M}_0$ 称为子模型链 $\mathcal{M}_\Gamma$ 的**最小元**，$\mathcal{M}_\gamma$ 称为子模型链 $\mathcal{M}_\Gamma$ 的**最大元**。

引理 4.1： 式 (4.1) 的子模型链中任意两个模型间均满足子模型的条件。

对式 (4.1) 中的任意两个模型 $\mathcal{M}_i$，$\mathcal{M}_j$ ，如果 $i < j$ ，由于模型链的长度为有限，容易验证 $\mathcal{M}_i$，$\mathcal{M}_j$ 满足定义 2.4 所确定的子模型的条件，即有 $\mathcal{M}_i \subseteq \mathcal{M}_j$ 。

可以用图的形式来表示子模型链的结构。图 4.1 表示了式(4.1)模型链的结构。

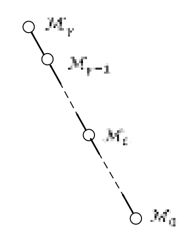


图 4.1　　子模型链结构

图 4.1 中，模型 $\mathcal{M}_0$ 处在模型链的最低层，模型 $\mathcal{M}_\gamma$ 处在模型链的最高层。图中表示出低层的模型是高层模型的子模型。

引理 4.2： 设 $\mathcal{M}_i$，$\mathcal{M}_j$ 是子模型链中的任意两个模型，如果 $i < j$，则有 $\mathcal{M}_i \cap \mathcal{M}_j = \mathcal{M}_i$。

根据子模型定义 2.4 和模型交运算定义 3.1，不难验证引理 4.2 成立。

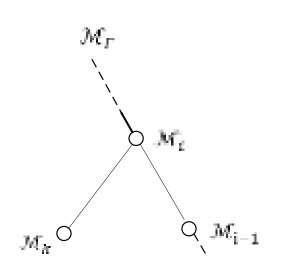
4.3 子模型链的树结构

从本节起，所论模型和子模型均在一阶语言 $\mathcal{L}=\{C, R, F\}$ 的模型定义范围内，为简单计，在提及模型和子模型时一般就不再明确指出一阶语言 $\mathcal{L}=\{C, R, F\}$ 了。

(1) 子模型链中的并结构

定义 4.3： 设 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma, \ldots, \mathcal{M}_i, \ldots, \mathcal{M}_1, \mathcal{M}_0\}$ 为一子模型链集，$\mathcal{M}_i \in \mathcal{M}_\Gamma\ (0 \le i \le \gamma)$。称 $\mathcal{M}_{i-1}$ 为 $\mathcal{M}_i\ (1 \le i \le \gamma)$ 的**直接子模型**。

设 $\mathcal{M}_k$ 为 $\mathcal{M}_\Gamma$ 外的另一模型，$\mathcal{M}_k \notin \mathcal{M}_\Gamma$。现考虑将 $\mathcal{M}_k$ 以受限并运算方式从 $\mathcal{M}_i$ 处接入该模型链中。如图 4.2 所示。



a. $\mathcal{M}_k$ 以受限并运算方式从 $\mathcal{M}_i$ 处接入　　*b.* $\mathcal{M}_k$ 与 $\mathcal{M}_i$ 组合为并模型 $\mathcal{M}_i'$

图 4.2　$\mathcal{M}_k$ 以受限并运算方式从 $\mathcal{M}_i$ 处接入模型链 $\mathcal{M}_\Gamma$ 中

在图 4.2 中，$\mathcal{M}_k$ 以受限并运算方式从 $\mathcal{M}_i$ 处接入模型链，由 $\mathcal{M}_k$ 与 $\mathcal{M}_i$ 组成并模型 $\mathcal{M}_i \cup_{sub} \mathcal{M}_k$，然后根据定理 4.1,将该节点模型修改为 $\mathcal{M}_i' = \mathcal{M}_i \cup_{sub} \mathcal{M}_k$，节点原来的模型 $\mathcal{M}_i$ 将不再独立存在于模型链中，只是在处理 $\mathcal{M}_i'$ 内部参数时，清楚关系 $\mathcal{M}_i' = \mathcal{M}_i \cup_{sub} \mathcal{M}_k$ 即可。

模型 $\mathcal{M}_k$ 以受限并运算方式从 $\mathcal{M}_i$ 处接入后，$\mathcal{M}_i$ 修改为 $\mathcal{M}_i'$，$\mathcal{M}_i'$ 的直接子模型变为 $\mathcal{M}_{i-1}$ 和 $\mathcal{M}_k$ 两个。

为了保证整个模型链 $\mathcal{M}_\Gamma$ 具有子模型的性质，还必须将 $\mathcal{M}_i$ 之上直至 $\mathcal{M}_\gamma$ 的所有模型都进行相应的扩充修改。定理 4.2 说明了这些修改的必要性。

定理 4.2： 设 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma, \ldots, \mathcal{M}_i, \ldots, \mathcal{M}_1, \mathcal{M}_0\}$ 为语言 $\mathcal{L}=\{C, R, F\}$ 的一子模型链，$\mathcal{M}_i \in \mathcal{M}_\Gamma\ (0 \le i \le \gamma)$。设 $\mathcal{M}_k$ 为 $\mathcal{M}_\Gamma$ 外的任意模型：$\mathcal{M}_k \notin \mathcal{M}_\Gamma$。如果将 $\mathcal{M}_k$ 以受限并运算方式从模型链的 $\mathcal{M}_i$ 节点接入，则必须对模型链中的 $\mathcal{M}_j\ (i \le j \le \gamma)$ 节点修改为：$\mathcal{M}_j' = \mathcal{M}_j \cup_{sub} \mathcal{M}_k$，以保持模型链的子模型特性不变。

证明： 在模型链 $\mathcal{M}_\Gamma$ 中，$\mathcal{M}_k = (M_k, C_k, R_k, F_k)$ 从第 i 节点接入，现考察模型链中的 $\mathcal{M}_j = (M_j, C_j, R_j, F_j)\ (i \le j \le \gamma)$ 节点情况，采用归纳法证明。

当 $j = i$ 时，由图 4.2 可看出，$\mathcal{M}_i' = (M_i', C_i', R_i', F_i') = \mathcal{M}_i \cup_{sub} \mathcal{M}_k$。根据定义 4.1，

可有：

a） 论域 $M_i \cup M_k = M_i'$；

b） $\mathcal{M}_i$，$\mathcal{M}_k$ 的并运算满足定义 3.2 中的模型受限并运算条件。

假设 $j = n-1$ 时，定理结论成立，即有 $\mathcal{M}_{n-1}' = \mathcal{M}_{n-1} \cup_{sub} \mathcal{M}_k$。

对 $j = n$ ，分析 $\mathcal{M}_k$ 接入前后的影响：

1） 在 $\mathcal{M}_k$ 接入前，对 $\mathcal{M}_\Gamma$ 中的 $j = n$ 节点和 $n-1$节点，有 $\mathcal{M}_{n-1} \subseteq \mathcal{M}_n$，根据子模型定义 2.4，以下结论成立：

a） 两个节点的论域 $M_{n-1} \subseteq M_n$；

b） 对任意常项符 $c \in \boldsymbol{\mathcal{L}}$，有 $c^{M_{n-1}} = c^{M_n}$；

c） 对任意 n−元关系符 $R \in \boldsymbol{\mathcal{L}}$，及任意 $a_1, \dots, a_n \in M_{n-1}$, 都有 $R^{M_{n-1}}(a_1, \dots, a_n)$ 当且仅当 $R^{M_n}(a_1, \dots, a_n)$；

d） 对任意 n−元函数符 $f \in \boldsymbol{\mathcal{L}}$，及任意 $a_1, \dots, a_n \in M_{n-1}$, 都有 $f^{M_{n-1}}(a_1, \dots, a_n) = f^{M_n}(a_1, \dots, a_n)$。

2) 在 $\mathcal{M}_k$ 接入后，对 $\mathcal{M}_\Gamma$ 中的 $j = n-1$ 节点，$\mathcal{M}_{n-1}' = \mathcal{M}_{n-1} \cup_{sub} \mathcal{M}_k$ 成立，此时，$\mathcal{M}_{n-1} \subseteq \mathcal{M}_n$成立，但 $\mathcal{M}_{n-1}' \subseteq \mathcal{M}_n$ 不一定成立。需对 $j = n$ 节点模型修改为 $\mathcal{M}_n'$ ：

a） 对论域，已有 $M_{n-1}' = M_{n-1} \cup M_k$。由 $M_{n-1} \subseteq M_n$ 可推出 $M_{n-1} \cup M_k \subseteq M_n \cup M_k$，令 $M_n' = M_n \cup M_k$，可得 $M_{n-1}' \subseteq M_n'$。

b) 对任意常项符 $c \in \boldsymbol{\mathcal{L}}$,根据引理 4.1,对 $n-1$ 节点,可有:如果 $c^{M_{n-1}} = c^{M_{n-1}'}$，则 $c \in C_{n-1}$ 且 $c \in C_{n-1}'$；如果 $c^{M_k} = c^{M_{n-1}'}$，则 $c \in C_k$ 且 $c \in C_{n-1}'$。对节点 n,由上面 1)的 b)，在 $\mathcal{M}_k$ 接入前，已有 $c^{M_{n-1}} = c^{M_n}$，故在 $\mathcal{M}_k$ 接入后，只需在节点 n 增加 $c \in C_k$ 的那部分内容，即增加 $c \in C_k$时，使得 $c^{M_k} = c^{M_n'}$。相当于对常项符，将 M_k 的那部分内容纳入 M_n' 中。

c） 对任意n−元关系符 $R \in \boldsymbol{\mathcal{L}}$,及任意 $a_1, \dots, a_n$,在n-1 节点,如果 $a_1, \dots, a_n \in M_{n-1}$，则有$R^{M_{n-1}}(a_1, \dots, a_n)$ 当且仅当 $R^{M_{n-1}'}(a_1, \dots, a_n)$；如果 $a_1, \dots, a_n \in M_k$，则有$R^{M_k}(a_1, \dots, a_n)$ 当且仅当 $R^{M_{n-1}'}(a_1, \dots, a_n)$。
而对于节点 n，由上面 1)的 c)，在 $\mathcal{M}_k$ 接入前，已有：当 $a_1, \dots, a_n \in M_{n-1}$时，$R^{M_{n-1}}(a_1, \dots, a_n)$ 当且仅当 $R^{M_n}(a_1, \dots, a_n)$当且仅当$R^{M_n'}(a_1, \dots, a_n)$，而在 $\mathcal{M}_k$ 接入后，需增加：如果 $a_1, \dots, a_n \in M_k$ 时，$R^{M_k}(a_1, \dots, a_n)$ 当且仅当 $R^{M_n'}(a_1, \dots, a_n)$。

d） 对任意 n−元函数符 $f \in \boldsymbol{\mathcal{L}}$，及任意 $a_1, \dots, a_n$，在 n-1 节点，如果 $a_1, \dots, a_n \in M_{n-1}$，则有 $f^{M_{n-1}}(a_1, \dots, a_n) = f^{M_{n-1}'}(a_1, \dots, a_n)$；如果 $a_1, \dots, a_n \in M_k$，则有 $f^{M_k}(a_1, \dots, a_n) = f^{M_{n-1}'}(a_1, \dots, a_n)$。对于节点 n，由上面 1)的 d)，在 $\mathcal{M}_k$ 接入前，已有：当 $a_1, \dots, a_n \in M_{n-1}$ 时，$f^{M_{n-1}}(a_1, \dots, a_n) = f^{M_n}(a_1, \dots, a_n) = f^{M_n'}(a_1, \dots, a_n)$，而在 $\mathcal{M}_k$ 接入后，需增加：如果 $a_1, \dots, a_n \in M_k$，则 $f^{M_k}(a_1, \dots, a_n) = f^{M_n'}(a_1, \dots, a_n)$。

综合以上情形，可得 $\mathcal{M}_n' = \mathcal{M}_n \cup_{sub} \mathcal{M}_k$，即有 $\mathcal{M}_j' = \mathcal{M}_j \cup_{sub} \mathcal{M}_k$。■

定义 4.4： 设 $\mathcal{M}_\Gamma = \{ \mathcal{M}_\gamma, \dots, \mathcal{M}_i, \dots, \mathcal{M}_1, \mathcal{M}_0 \}$ 为语言 $\mathcal{L}$ ={C, R, F}的一子模型链，

$\mathcal{M}_\mathrm{i} \in \mathcal{M}_\Gamma\,(\,0 \le \mathrm{i} \le \gamma\,)$。设 $\mathcal{M}_\mathrm{k}$ 为 $\mathcal{M}_\Gamma$ 外的任意模型，$\mathcal{M}_\mathrm{k} \notin \mathcal{M}_\Gamma$。如果 $\mathcal{M}_\mathrm{k}$ 从子模型链 $\mathcal{M}_\Gamma$ 中的某一节点 $\mathcal{M}_\mathrm{i}$ 处接入 $\mathcal{M}_\Gamma$，且满足定理 4.2 的条件，则称模型 $\mathcal{M}_\mathrm{k}$**可链接入**子模型链 $\mathcal{M}_\Gamma$。

定理 4.3： 任意模型 $\mathcal{M}_\mathrm{k}$ 均可链接入到一个子模型链中。

定理 4.3 的结论是显然的，它揭示了这样一个事实：任何一阶语言的模型只要采用模型受限并运算方式，均可链接入其它模型链中，且能保证运算数据的完整性，使模型能正常工作。当然，在实际应用中，模型的组合应当根据应用需要，引入必需的模型；同时，在引入新模型时还应当根据实际情况对模型内部参数做适当调整，以确保组合模型的性能，这里仅是揭示了模型的这种组合的可能性。

(2) 子模型链中并结构的扩展

在 4.3.1 的基础上，子模型链中的并结构方式可以进一步拓展：一是在模型链的多个节点处以受限并运算方式接入不在链中的其它模型；二是在同一节点处以受限并运算方式同时接入多个其它模型。现分别讨论于后。

1) 在模型链中多个节点处接入其它模型

设 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma，\cdots，\mathcal{M}_j，\cdots，\mathcal{M}_i，\cdots，\mathcal{M}_0\}$ 为一子模型链集，$\mathcal{M}_p$，$\mathcal{M}_q$ 为两个不在模型链中的其它模型。现考虑以受限并运算方式将 $\mathcal{M}_p$，$\mathcal{M}_q$ 分别从节点 $\mathcal{M}_i$，$\mathcal{M}_j$ 处接入模型链，如图 4.3 所示。

图中，模型 $\mathcal{M}_p \notin \mathcal{M}_\Gamma$，$\mathcal{M}_q \notin \mathcal{M}_\Gamma$。由于它们是以受限并运算方式分别从节点 $\mathcal{M}_i$，$\mathcal{M}_j$ 处接入模型链的，由定理 4.2，它们接入后对关系和函数分别都是各自封闭的，

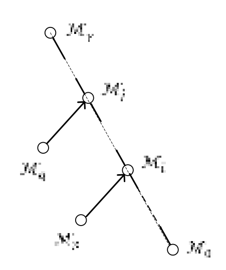


图 4.3　　模型链的多个节点接入其它模型

因此可以分别考虑其对模型链特性的影响。根据定理 4.2，有以下推论：

推论4.3： 设 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma，\cdots，\mathcal{M}_j，\cdots，\mathcal{M}_i，\cdots，\mathcal{M}_0\}$ 为一子模型链集，$\mathcal{M}_p$，$\mathcal{M}_q$ 为两个不在模型链中的其它模型。若将 $\mathcal{M}_p$，$\mathcal{M}_q$ 以受限并运算方式分别从 $\mathcal{M}_i$，$\mathcal{M}_j(i<j)$ 接入模型链，则对模型链中的 $\mathcal{M}_K$ 节点，需按下列方式将模型参数由 $\mathcal{M}_K$ 修改为 $\mathcal{M}_k'$，以保持模型链的子模型特性不变：

a) $i \le k < j$，$\mathcal{M}_k' = \mathcal{M}_K \cup_{sub} \mathcal{M}_p$，

b) $j \le k$，$\mathcal{M}_k' = \mathcal{M}_K \cup_{sub} \mathcal{M}_p \cup_{sub} \mathcal{M}_q$。

推论 4.3 有类似于叠加原理的特性。

2) 在模型链中的同一节点处接入多个其它模型

设 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma，\cdots，\mathcal{M}_k，\cdots，\mathcal{M}_0\}$ 为一子模型链集，$\mathcal{M}_{p1}，\cdots，\mathcal{M}_{pm}$ 为不在 $\mathcal{M}_\Gamma$ 中的另一组模型。现考虑从 $\mathcal{M}_\Gamma$ 中的节点 $\mathcal{M}_k\,(\,0 \le k \le \gamma\,)$ 处同时接入该组模型，如图 4.4 所示。

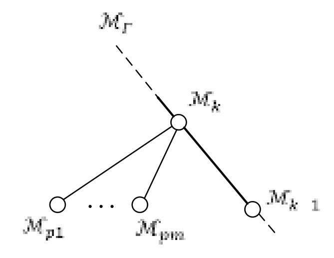


图 4.4 多个模型从模型链一节点处接入

如图 4.4 所示，多个模型 $\mathcal{M}_{p1}$，⋯，$\mathcal{M}_{pm}$ 从模型链 $\mathcal{M}_\Gamma$ 的节点 $\mathcal{M}_k$ 处以受限并运算方式接入，现考察其对模型链的影响。

由于模型 $\mathcal{M}_{p1}$，⋯，$\mathcal{M}_{pm}$ 是以受限并运算方式接入的，其特性是对关系和函数都分别封闭，故可将该组模型进行并运算 $\mathcal{M}'_p = \mathcal{M}_{p1} \cup_{sub} \cdots \cup_{sub} \mathcal{M}_{pm}$，易看出下面的推论成立。

推论4.4： 设 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma, \cdots, \mathcal{M}_k, \cdots, \mathcal{M}_0\}$ 为一子模型链集，$\mathcal{M}_{p1}$，⋯，$\mathcal{M}_{pm}$ 为不在 $\mathcal{M}_\Gamma$ 中的另一组模型。现从 $\mathcal{M}_\Gamma$ 中的节点 $\mathcal{M}_k$（$0 \le k \le \gamma$）处同时以受限并运算方式接入该组模型，则模型链 $\mathcal{M}_\Gamma$ 中的节点 $\mathcal{M}_k$（$0 \le k \le \gamma$）及以上节点模型需作如下修改，以保持子模型链的特性不变：

a） 对 $\mathcal{M}_k$（$0 \le k \le \gamma$）节点，$\mathcal{M}'_k = \mathcal{M}_k \cup_{sub} \mathcal{M}'_p$，其中 $\mathcal{M}_k$ 为接入前模型，$\mathcal{M}'_p = \mathcal{M}_{p1} \cup_{sub} \cdots \cup_{sub} \mathcal{M}_{pm}$。

b） 对 $\mathcal{M}_k$ 以上节点 $\mathcal{M}_l$ $(k < l \le \gamma)$，$\mathcal{M}'_l = \mathcal{M}_l \cup_{sub} \mathcal{M}'_p$，$\mathcal{M}_l$为接入前模型。

3） 一种生成子模型链的方法

根据以上对模型链扩展的分析，我们可以通过模型扩展来生成子模型链。

设 $\mathcal{M}^0_\Gamma = \{\mathcal{M}_\gamma, \cdots, \mathcal{M}_i, \cdots, \mathcal{M}_0\}$为语言 $\boldsymbol{\mathcal{L}} = \{C, R, F\}$ 的一个非子模型链的模型集。我们可以按以下步骤来生成一个子模型链：

a） 从 $\mathcal{M}^0_\Gamma$ 中任选出一个模型，比如模型 $\mathcal{M}_i$ $(0 \le i \le \gamma)$，$\mathcal{M}_\Gamma \Leftarrow \mathcal{M}^0_\Gamma - \{\mathcal{M}_i\}$，构造模型链 $\mathcal{M}'_\gamma \Leftarrow \mathcal{M}_i$，$\mathcal{M}'_\Gamma = \{\mathcal{M}'_\gamma\}$;

b） 从 $\mathcal{M}_\Gamma$ 中任选出一模型 $\mathcal{M}_j$，$\mathcal{M}_\Gamma \Leftarrow \mathcal{M}_\Gamma - \{\mathcal{M}_j\}$;

c） 将 $\mathcal{M}_j$ 链接到 $\mathcal{M}'_\Gamma$ 上：修改 $\mathcal{M}'_\Gamma$，$\forall \mathcal{M}_k \in \mathcal{M}'_\Gamma$，$\mathcal{M}_k = \mathcal{M}_k \cup_{sub} \mathcal{M}_j$，再链接 $\mathcal{M}'_\Gamma = \mathcal{M}'_\Gamma \cup \{\mathcal{M}_j\}$;

d) 检查 $\mathcal{M}_\Gamma$ 是否为空，若不为空，则重复至 b)；
为空，则结束退出。

容易验证以上子模型链生成法是可行的。可有以下定理：

定理 4.4： 对语言 $\boldsymbol{\mathcal{L}} = \{C, R, F\}$的任意模型集 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma, \ldots, \mathcal{M}_i, \ldots, \mathcal{M}_0\}$，可以通过模型扩展来生成子模型链：$\mathcal{M}'_\gamma \supseteq \cdots \supseteq \mathcal{M}'_i \supseteq \cdots \supseteq \mathcal{M}'_0$。

(3) 以子模型链为基础的树结构

以上讨论都是在一条模型链上进行的。对子模型链集 $\mathcal{M}_\Gamma = \{\mathcal{M}_\gamma, \cdots, \mathcal{M}_j, \cdots, \mathcal{M}_i, \cdots, \mathcal{M}_0\}$，我们研究了在其中的任意节点处以受限并运算方式接入其它模型，即在子模型链 $\mathcal{M}_\gamma \supseteq \cdots \supseteq \mathcal{M}_j \supseteq \cdots \supseteq \mathcal{M}_i \supseteq \cdots \supseteq \mathcal{M}_0$ 上链接上其它模型，形成若干分叉。如果链接的模

型又属于另一子模型链集，比如对于子模型链集 $\mathcal{M}_{\mathrm{K}}=\{\mathcal{M}_{km}，\cdots，\mathcal{M}_{ki}，\cdots，\mathcal{M}_{k0}\}$，将子模型链 $\mathcal{M}_{km}\supseteq\cdots\supseteq\mathcal{M}_{ki}\supseteq\cdots\supseteq\mathcal{M}_{k0}$ 中的子模型 $\mathcal{M}_{ki}$ 接入 $\mathcal{M}_{\Gamma}$ 中的 $\mathcal{M}_j$ 节点，实质上就相对于将 $\mathcal{M}_K$ 的一部分 $\mathcal{M}_{ki}\supseteq\cdots\supseteq\mathcal{M}_{k0}$ 接入 $\mathcal{M}_{\Gamma}$，形成了一个新的模型结构。

定义 4.5：在模型链 $\mathcal{M}_{\Gamma}$： $\mathcal{M}_{\gamma}\supseteq\cdots\supseteq\mathcal{M}_{i+1}\supseteq\mathcal{M}_i\supseteq\mathcal{M}_{\mathrm{i-1}}\supseteq\cdots\supseteq\mathcal{M}_0$ 中，$\mathcal{M}_i$ 为其中某一节点，如果对模型链中另一节点 $\mathcal{M}_q$ 有 $\mathcal{M}_i\supseteq\mathcal{M}_q$，则称 $\mathcal{M}_q$ 为 $\mathcal{M}_i$ 的模型链后继节点，或简称**后继**，若不存在 $\mathcal{M}_k\in\mathcal{M}_{\Gamma}$，使得 $\mathcal{M}_i\supseteq\mathcal{M}_k\supseteq\mathcal{M}_q$，则称$\mathcal{M}_q$ 为 $\mathcal{M}_i$ 的**直接后继**；如果 $\mathcal{M}_q\supseteq\mathcal{M}_i$，则称 $\mathcal{M}_q$ 为 $\mathcal{M}_i$ 的模型链前驱节点，或简称**前驱**，若不存在 $\mathcal{M}_k\in\mathcal{M}_{\Gamma}$，使得 $\mathcal{M}_q\supseteq\mathcal{M}_k\supseteq\mathcal{M}_i$，则称 $\mathcal{M}_q$ 为 $\mathcal{M}_i$ 的**直接前驱**。

由定义 4.4 和前面有关在子模型链 $\mathcal{M}_{\Gamma}$中接入其它模型的论述，不难得出以下结论：

推论4. 5：设模型链 $\mathcal{M}_{\Gamma}$：$\mathcal{M}_{\gamma}\supseteq\cdots\supseteq\mathcal{M}_i\supseteq\cdots\supseteq\mathcal{M}_0$ 和 $\mathcal{M}_{\mathrm{K}}$：$\mathcal{M}_{km}\supseteq\cdots\supseteq\mathcal{M}_{ki}\supseteq\cdots\supseteq\mathcal{M}_{k0}$ 均为子模型链。如果将模型 $\mathcal{M}_{\mathrm{K}}$ 的子模型 $\mathcal{M}_{ki}$ 从 $\mathcal{M}_i$ 处接入模型链 $\mathcal{M}_{\Gamma}$ 中，则相当于将 $\mathcal{M}_{ki}$ 及其后继接入了 $\mathcal{M}_{\Gamma}$，即将子模型链 $\mathcal{M}_{ki}\supseteq\cdots\supseteq\mathcal{M}_{k0}$ 接入了模型链 $\mathcal{M}_{\Gamma}$ 中；而在接入后，需将 $\mathcal{M}_{\Gamma}$ 的接点 $\mathcal{M}_i$及前驱进行修改：$\mathcal{M}_j'=\mathcal{M}_j\cup_{sub}\mathcal{M}_{ki}$（$i\le j\le\gamma,\mathcal{M}_j$ 为接入前的模型参数）以保持 $\mathcal{M}_{\Gamma}$ 的子模型链特性不变。

由此，我们可以构造一种由子模型链接形成的新型子模型结构。

定义 4.6：设 $\mathcal{M}_{\Gamma}$：$\mathcal{M}_{\gamma}\supseteq\ldots\supseteq\mathcal{M}_i\supseteq\ldots\supseteq\mathcal{M}_0$ 为一子模型链，$\{\mathcal{M}_{K1},\ldots,\mathcal{M}_{Km}\}$ 为一子模型链集。现将该子模型链集中任一子模型链 $\mathcal{M}_{Kj}(1\le\mathrm{j}\le\mathrm{m})$接入子模型链 $\mathcal{M}_{\Gamma}$ 的某一子模型 $\mathcal{M}_i$ 处，即将 $\mathcal{M}_{Kj}$ 的最大元链接到 $\mathcal{M}_i$ 处。依照这一链接方法，还可将该子模型链集中的其它子模型链接入 $\mathcal{M}_{\Gamma}$ 中。这样，便可形成一种模型链的树形结构，我们称之为**子模型树**。

图 4.5 表示了一种子模型树结构。其中，由一系列模型节点构成子模型树集合 $\mathcal{M}_{\Gamma}=\{\mathcal{M}_{\gamma},\ldots,\mathcal{M}_j,\ldots,\mathcal{M}_i,\ldots,\mathcal{M}_p,\ldots,\mathcal{M}_0\}$。其中，$\mathcal{M}_{\gamma}$ 为树的根节点，即最高节点；$\mathcal{M}_j$，$\mathcal{M}_i$，$\mathcal{M}_p$，... 等为树的中间节点；$\mathcal{M}_0$，$\mathcal{M}_{01}$，... 等为树的叶节点，即末端节点。从最高节点 $\mathcal{M}_{\gamma}$ 到某一末端节点(比如 $\mathcal{M}_{01}$)，如果存在一条通路，则这一条通路上的所有节点即构成一条子模型链。如图所示，

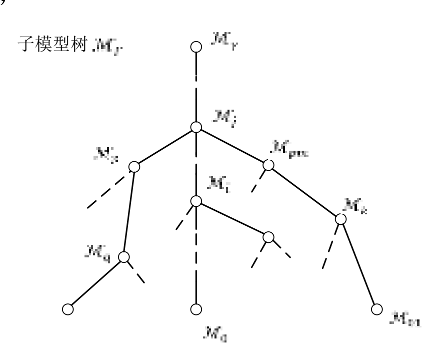


图 4.5 由子模型链接形成的模型树

$\mathcal{M}_{\gamma},\ldots,\mathcal{M}_{\mathrm{j}}$，$\mathcal{M}_{pm}$，$\mathcal{M}_k$，$\mathcal{M}_{01}$ 在根节点 $\mathcal{M}_{\gamma}$ 和末端节点 $\mathcal{M}_{01}$ 间构成一条完整通路，则这些节点模型就构成一条子模型链：$\mathcal{M}_{\gamma}\supseteq\cdots\supseteq\mathcal{M}_j\supseteq\mathcal{M}_{pm}\supseteq\mathcal{M}_k\supseteq\mathcal{M}_{01}$。实际上，根据定义

4.4，该模型链中除了根节点和末端节点外，模型链中的相邻节点在模型树中均互为直接前驱或直接后继。于是有：

推论4.6： 在一个子模型树中，任一末端节点与从它出发到根节点之间的所有直接前驱节点就组成一条完全子模型链；或反过来，从根节点到任一末端节点之间的所有直接后继节点就组成一条完全子模型链。一个子模型树具有的完全子模型链数与其末端节点数相同。

子模型树可以清楚表示出各个模型之间的逻辑关系，因此在利用模型作数据处理时就会非常方便。例如我们在用某一模型进行数据处理时，如需通过引用它的子模型来处理某些数据，即可借助子模型树查找到它的子模型，逻辑上可以直接调用该子模型。这为具体的系统模型实现提供了方便。

4.4 以子模型链为基础的模型网络结构

子模型树描述了一个大的研究对象的模型和与之有关联的其它模型之间的逻辑关系。子模型树的方法可以推广到更多对象的情况。实际上，一些基础的模型方法可以用于多种更高层次模型的研究，也就是说，某个基础的模型成果可以用于多种不同的更高层次对象的研究。这是必要的，也是很有吸引力的。依循子模型树的构造方法，我们可以针对不同的对象构造其子模型链，形成子模型树，然后再按相同子模型进行关联合并，形成一种网络结构；或者反过来，从底层模型出发，链接更高层次的模型，最后到达作为目标对象的模型，形成**子模型网络**结构。

图 4.6 表示了一种以子模型树为基础的子模型网络结构。

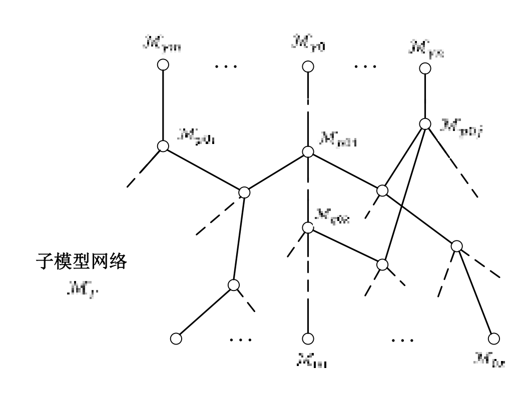


图 4.6 一种子模型网络结构

子模型网络结构也是以子模型链为基础建立的。根据推论 4.4，从任一根节点出发，找出它的一个直接后继节点，再找出下一个直接后继节点，…,直至某一末端节点，所找出的全部节点就构成一条子模型链；如果某个中间节点有不止一个直接后继，则又可再派生出另一条包含该中间节点的子模型链……；当穷尽了所有可能的子模型链后，即找出了一个根节点所关联的全部子模型链。对网络结构的每一根节点均可重复上述过程，找出其关联的全部子模型链。有以下推论：

推论4.7： 在子模型网络 $\mathcal{M}_\Gamma$中，设其根节点为 $\mathcal{M}_{\Gamma 1}, \ldots, \mathcal{M}_{\Gamma i}, \ldots,\ \mathcal{M}_{\Gamma m}$，每个根节点关联的完全子模型链数为 n_i，则 $\mathcal{M}_\Gamma$ 所关联的完全子模型链数为 $N_\Gamma = \sum_i n_i$。对任意子模型网络，设它的根节点数为 m，末端节点数为 k，则它所关联的完全子模型链数最小值为 m，

最大值为 mk。

子模型网络反映了不同对象与所关联的其它模型的逻辑关系。一般说来，底层模型的前驱节点多于高层模型的前驱节点，这也是符合实际情况的。与高层模型相比，底层模型的方法会更简单些，将被更多更复杂的模型所使用。

5. 子模型方法应用展望

本文所涉及的模型结构处理方法是以子模型为基础的模型组合方法，我们暂称其为模型的子模型组合方法，简称**子模型方法**。

5.1 子模型方法应用的基础条件

子模型方法主要是解决了由小模型组建大模型的可能性问题，定理 4.3 对此作出了断言。但一般而言，采用模型组合方法的目的是为了使模型之间能够相互利用处理的数据结果，提高组合模型的性能。因此，在具体应用中，模型的组合方案应根据实际应用的需要来制定。一个模型要能够利用其它模型处理数据的结果，则必须要共享其它模型处理后的数据，实现模型间数据传递。从模型论的角度看，能够相互传递数据的模型之间的数据域一定存在交集，故有以下引理：

引理5.1： 设 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$ 和 $\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$ 是一阶语言 $\boldsymbol{\mathcal{L}}=\{C, R, F\}$ 的两个模型。如果要在 $\mathcal{M}_1$ 和 $\mathcal{M}_2$ 之间实现数据传递，其必要条件是它们的论域 M_1 和 M_2 之交集一定非空，即有 $M_1 \cap M_2 \neq \Phi$。

由引理 5.1，对于两个模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$ 和 $\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$，其模型的组合存在交、并运算两种情况，对其论域的要求分析如下：

1） **交运算：**$\mathcal{M}_0 = \mathcal{M}_1 \cap \mathcal{M}_2 = (M_0, C_0, R_0, F_0)$为二者的交运算。如果 $\mathcal{M}_1 \cap \mathcal{M}_2 \neq \Phi$，即交运算 $\mathcal{M}_0$ 成立，依据定义 3.1，必有 $M_1 \cap M_2 \neq \Phi$。

2） **并运算：**本文只限于研究模型受限并运算的情况，即只考虑 $\mathcal{M}_0 = \mathcal{M}_1 \cup_{sub} \mathcal{M}_2 = (M_0, C_0, R_0, F_0)$。针对并模型 $\mathcal{M}_0$ 的论域 $M_0 = M_1 \cup M_2$，可有以下两种情况：

   a） $M_1 \cap M_2 = \Phi$。在模型受限并运算的前提下，模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 之间数据无交集，数据无传递。虽然运算 $\mathcal{M}_0 = \mathcal{M}_1 \cup_{sub} \mathcal{M}_2$ 成立，但组合模型的性能未有改变和提高，只是二者的拼接。这种情况，在大的模型组合中应引起注意。

   b） $M_1 \cap M_2 \neq \Phi$。这种情况能实现模型间的数据传递和相互利用，组合后的模型性能会有所改变和提高，因此，在模型组合时应充分考虑和利用模型具有的这种特点，制定模型组合方案。

由此可看出，模型的论域直接影响到模型的组合方案和性能，是建立组合模型的基础。

5.2 关于模型受限并运算的拓展

本文提出的模型受限并运算是针对模型论域而进行的，是一个很强的约束条件。一般而言，对一阶语言 $\boldsymbol{\mathcal{L}}=\{C, R, F\}$ 的两个模型 $\mathcal{M}_1 = (M_1, C_1, R_1, F_1)$ 和 $\mathcal{M}_2 = (M_2, C_2, R_2, F_2)$，它们的论域之间的交集存在以下三种情况：

1） M_1 和 M_2 完全重叠，即有 $M_1 = M_2$；

2） M_1 和 M_2 部分重叠，即有 $M_1 \cap M_2 \neq \Phi$，且 $M_1 \neq M_2$；

3） M_1 和 M_2 无重叠，$M_1 \cap M_2 = \Phi$。

对 1)，$\mathcal{M}_1$ 与 $\mathcal{M}_2$ 之间的并运算与受限并运算效果完全相同。对 2)，$\mathcal{M}_1$ 与 $\mathcal{M}_2$ 之间的并运算应采用受限并运算，模型之间存在数据传递。对 3)，$\mathcal{M}_1$ 与 $\mathcal{M}_2$ 之间的并运算应采用受限并运算，模型之间不存在数据传递。对于情况 2)和 3)，能否在进行并运算时放宽本文提出的受限并运算约束条件，使并运算能在更大范围内成立；或反过来修改模型，使并运算在更大范围内成立。这是一个很有意义的课题，在探究模型组合技术时，应进一步深入研究。

5.3 子模型方法应用展望

针对目前人工智能领域的研究情况，对子模型方法的应用提出以下意见和建议：

1、 在模型中引入其它成熟模型

根据以上对子模型树和子模型网络的分析可以看出，能够在一个明确逻辑关系的模型组合中引入其它模型，以增加模型组合的性能和能力。如果被引入的是一个成熟的模型，已经在其它地方验证了该模型具有优良的性能，不必在新模型组合中对其重新开发，而只要对其使用环境参数做少量适应性调整即可嵌入到新模型中，这样可以减少很多重复研发工作量。显然，这对智能系统的研发是很有意义的。

2、 关于模型分解

根据以上对子模型树和子模型网络的分析可以看出，一个规模较大的模型可以由一系列较小的子模型逐级组成。因此，存在这种可能：一个大型的模型任务，可以按照子模型原则将其划分为较小的模型来实现。这样，可以在研发过程中将原始模型任务分解成若干较小的子模型分别进行研发，待各子模型研发完成后，再进行组合。这样，可以充分发挥研发团队的作用，并行工作；同时还存在引入成熟模型的可能，故可以加快研发进度，缩短研发周期。

3、 关于不同体制模型的混用

在以上子模型树和子模型网络的论述中，所提及的模型和子模型都是数理逻辑模型论中的模型，不是一般的具体数学模型，而是一种结构模型，是一般具体模型的抽象。而在处理具体实际对象问题时，必须使用具体模型，包括数学模型。但模型论中的模型之间所确定的逻辑原则，在具体对象模型中应该是适用的。

在具体对象的数学模型中，一般使用一整套数学公式去描述和刻画具体对象，当然其中也只是抓住了具体对象的主要特征进行描述，忽略了许多次要因素，但最终要能够通过模型对具体对象的行为进行认知、验证、预测和复现。通过模型，人们可以对具体对象的行为有深刻的认识，可以主动地改进模型的性能和对具体对象的处理。

在当下的人工智能领域，特别是深度学习网络，已经取得瞩目的成效。但该领域建立的模型一般是网络模型，通过调整模型的内部参数去提升模型的性能。这类模型也是一种具体对象模型，但除了模型中部分采用了明确的数学函数（如 relu，softmax 等函数）外，内部并无明显的数学公式刻画。如果把刻画一个具体对象的这种模型看成为一个整体，从外部看，它具有人们需要的功能，但又未使用具体的数学公式去描述它，只能将其看成为具有特定功

能的黑箱[9][10]。但在当下，这种模型的性能优越，已得到广泛应用。

我们关心的问题是，一个大的模型系统中，能否同时包含以上两类模型？从以上对子模型树和子模型网络的分析可看出，答案应该是肯定的。因此，将两类模型混用，组合成新的更大的具体模型系统是可能的，这值得进一步深入研究。

4、 子模型方法拓展到一般模型方法

本文第四节的讨论中，均是采用子模型方法进行的，其基础是定义 3.2 确定的模型受限并运算规则。其中，核心就是对任意 n-元变量取值组合 $a_1, \dots, a_i, \dots, a_n$ 的取值范围作出了限定，以保证函数运算的封闭，进而使并运算能够成立。如果不排除

$\{(a_1, \dots, a_i, \dots, a_j, \dots, a_n) | a_i, a_j \in M_0, \exists (a_i \in M_1 \text{ 且 } a_j \in M_2 \text{ 且 } a_j \notin M_1 \cap M_2)\}$的情形，

则一般会出现函数不封闭，造成并运算不成立，于是我们对运算进行了限定。一种可能的问题是：是否存在某种并运算，既不排除上述 n-元变量取值组合的情形，又能使运算中的函数封闭？答案似乎是肯定的。为使这种并运算成立，一个办法是找出另一种使函数封闭的并运算（非本文的受限并运算），这在某种特定条件下是可能的；另一个办法是通过验证方法确认在该并运算后函数是封闭的。5.2 节已从论域角度提出了这种可能性。

5、 关于子模型网络优化问题

一个由若干子模型组成的大型模型网络亦存在优化问题。当前，人们处理的模型优化一般都是针对单个模型进行的。对于有明确函数关系的数学模型，已有很多成熟的优化方法；对于没有明确函数关系的神经网络，亦有很多通过调节模型参数的调优方法。而对于由若干子模型组成的大型模型网络来说，调优问题要复杂得多。按当前所采用的优化方法，我们可以把每个子模型单独优化，但模型组合起来后不一定是整体优化的。如果要作整体优化，则需考虑子模型网络的实际情况。因子模型网络中的模型关系是一种结构逻辑关系，不是数学函数关系，直接使用当前的优化方法来对子模型网络进行优化是行不通的，需根据子模型网络的特点来制定优化方案。这里提出几点意见，供进一步研究时参考。

设有子模型链 $\mathcal{M}_\Gamma = \mathcal{M}_\gamma \supseteq \cdots \supseteq \mathcal{M}_j \supseteq \mathcal{M}_i \supseteq \cdots \supseteq \mathcal{M}_0$。其中，每个子模型都有可调整的参数，可以通过调节这些参数来优化模型的性能，这在神经网络中是常用的方法。在整个模型链中，低层模型的参数调节会影响到高层模型的响应，它是通过低层模型整体去影响高层模型的；而对高层模型，还有自身可调节的参数。因此，我们可以对每个子模型引入**独立参数**的办法，即：每个子模型中可具有独自进行调节的参数。例如，在上述模型链中，$\mathcal{M}_j$，$\mathcal{M}_i$ 是相邻的两个节点，$\mathcal{M}_j \supseteq \mathcal{M}_i$。设 $\mathcal{M}_i$ 具有独立参数 $a_{i0}, a_{i1}, \dots, a_{ip}$，$\mathcal{M}_j$具有独立参数 $a_{j0}, a_{j1}, \dots, a_{jq}$，这些独立调节参数是每个子模型在单独优化时要进行调节的。我们可以有两种办法来对子模型链调优。一种方法是子模型链建立后，针对它的所有独立参数进行一次性整体优化，这与我们当前采用的单个模型调优方法类似。另一种方法是，不再针对全部模型参数调优，而是将每个子模型的独立参数乘以一个加权系数 k_i，各子模型单独优化时，可暂时令各 k_i=1，而在模型链建立后进行整体调优时，我们只针对这些系数进行调优。例如，对上述的 $\mathcal{M}_i$，$\mathcal{M}_j$，其独立参数分别设为 $k_i a_{i0}, k_i a_{i1}, \dots, k_i a_{ip}$ 和 $k_j a_{j0}, k_j a_{j1}, \dots, k_j a_{jq}$，在

$\mathcal{M}_i$，$\mathcal{M}_j$ 单独进行优化时，可暂设 $k_i = 1, k_j = 1$，对 $a_{i0}, a_{i1}, \dots, a_{ip}$ 和 $a_{j0}, a_{j1}, \dots, a_{jq}$ 进行优化，在整个子模型链建立后，再通过调节 k_i, k_j，⋯ 这些系数进行整体优化。这样调节的参数在数量上就少多了。当然，这里只是提出一个方向，具体的子模型链与子模型树如何进行优化更好，需进行深入研究。

6、 关于子模型网络的应用

当前，多模态人工智能[11][12]和多智能体系统[13][14][15][16]是人工智能领域的重要发展方向，已取得瞩目的成绩。这两类系统都涉及多个模型间的数据转换、数据交互和相互协同问题。而子模型网络从逻辑结构上体现了不同模型间的逻辑关系，因此，子模型网络的深入研究对多模态系统和多智能体系统的技术支持有一定的现实意义。本文只是在子模型链的基础上提出了子模型网络的设想，未及深入讨论。根据实际需要，对这一课题做进一步的深入研究很有必要。

7、 模型的学习与进化

本文所提出通过构建子模型链和子模型树的方法，实质是体现了通过小模型组合成大模型的模型构建思路。也就是通过较小模型的建立、验证、优化，使其成熟，然后在模型系统中逐步增加成熟模型，使模型的体量越来越大，功能越来越强。这个过程有点类似于某种“学习”过程。只不过现在人们在构建大的模型系统时，这种学习过程是由人工实现的。如果我们在智能体的研发中，利用数理逻辑中的知识表达和知识推理，能够赋予智能体某种自身可以发现、建立、验证、优化较小的模型，或者智能体具有从人工建立的模型库中获取成熟的模型，然后添加到自身系统中去的能力，那么，该智能体就具有某种程度的自学习功能了。这是一个很具吸引力的研究方向，通过子模型链和子模型树的方法，使我们看到了一种新的可能路径。

6. 结语

当前，在人工智能领域，主流方向是将一个系统作为一个模型整体来进行研究。特别是在神经网络的研究中，不管系统大与小，几乎都是将研究对象作为一个整体进行规划组建，包括模型的构成、参数的设置、模型训练与调优，都是将整个系统作为一个模型来处理的。与主流方向不同的是，能否在新的研发中，直接利用一些已经经过验证的成熟成果来组建新系统，使新系统逐步增大，功能逐步增强，这是智能体系统一种新的实现途径。本文正是在这一研究方向上的一种尝试。

文中已经指出，不受限的模型并运算体现了从形式语言的语法角度去表达的思想，而受限模型并运算则是侧重于从语义角度去表达和实现，更接近于具体模型的研究。本文从数理逻辑模型论角度，提出了模型受限并运算的方法，实现了模型在一定约束条件下的并运算。在此基础上，论述了子模型链、子模型树以及子模型网络的组建原理，论述了由小模型组建成大规模模型的可行方法。本文的研究取得了两方面的成效：一是对智能体装置的开发和实现给出了新的思路和途径；二是在数理逻辑模型论的研究上提出了一个新的研究方向，即从模型论角度去研究人工智能问题。众所周知，数理逻辑是计算机的理论基础，在知识表达和

逻辑推理方面有独特的优势。从模型论角度去研究人工智能，当下人工智能领域的热门课题，诸如知识的表达，知识的学习与进步，智能体的进化等都可以在数理逻辑的理论框架下去研究，去实现。这对人工智能理论与技术、数理逻辑理论本身的发展来说，都是很有意义的。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit U, Jones L,Gomez A N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, USA: Curran Associates Inc., 2017. 6000–6010.

[2] Devlin J, Chang M W, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics:Human Language Technologies. Minneapolis, Minnesota, USA: Association for Computational Linguistics, 2018. 4171−4186.

[3] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, Florencia B, et al. GPT-4 technical report. arXiv preprint arXiv: 2303. 08774, 2023.

[4] 孟小峰等，科学发现中的机器学习方法研究. 计算机学报，2023，第 5 期，877-895

[5] 赵希顺，简明数理逻辑，科学出版社，2021.11.

[6] 罗里波，模型论及其在计算机科学中的应用，北京师范大学出版社，2012.1.

[7] Michael Huth, Mark Ryan，Lojic in Computer Science ：Modelling and Reasoning about Systems，Second Edition，ISBN 987-7-111-21397-0.

[8] 孙希文，数理逻辑，高等教育出版社，2019.11.

[9] Hutson M. Artificial intelligence faces reproducibility crisis. Science, 2018, 359 (6377): 725-726

[10] Khan S, Naseer M, Hayat M, Zamir S W, Khan, F S, Shah M.Transformers in vision: A survey. ar Xiv preprint arXiv: 2101.01169, 2021.

[11] RIK K，DHANUSH BEKAL，YI L，et al. Text Generation from Knowledge Graphs with Graph Transformers［C］// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics：Human Language Technolo- gies. Minnesota：Association for Computational Linguis- tics，2019：2284-2293.

[12] Selva J, Johansen A S, Escalera S, Nasrollahi K, Moeslund TB, Clapés A. Video transformers: A survey. arXiv preprint arXiv: 2201.05991, 2022.

[13] Wang F Y, Miao Q H, Li L X, Ni Q H, Li X, Li J J, et al. When does sora show: The beginning of TAO to imaginative intelligence and scenarios engineering. IEEE/CAA Journal of Automatica Sinica, 2024, 11(4): 809−815

[14] ZHAO W S, QUERALTA J P, WESTERLUND T. Sim-to-real transfer in deep reinforcement learning for robotics:a survey[C]//Proceedings of 2020 IEEE Symposium Series on Computational Intelligence.[S. l.]:IEEE, 2020:737-744.

[15] LOWE R, WU Y I, TAMAR A, et al.Multi-agent actor-critic for mixed cooperative-competitive environ ments[J].Advances in Neural Information Processing Systems, 2017, 30: 6382-6393.

[16] IQBAL S, SHA F.Actor attention critic for multi-agent reinforcement learning[C]//Proceedings of the 36th International Conference on Machine Learning. [S.l.:s.n.], 2019:5261-5274.

(通讯作者：刘志模 E-mail: hylj2010@sohu.com)

作者贡献声明：

刘志模：提出研究思路，论文起草，终版审定。

陈　振：审改论文，与实体模型进行了对比，对应用前景进行了分析并提出审改意见。

王晓山：审改论文，对论文的应用前景等提出审改意见。

刘　徽：审改论文，对论文的理论论述进行了审改，对应用前景提出审改意见。